Modelli DPM per problemi di clustering

Gaia Addis (864410), Lucrezia Tusetti (864790)

17 Dicembre 2024

Indice

1. Introduzione al problema

Il fine di questo elaborato è quello di affrontare il problema del clustering attraverso metodi alternativi rispetto ai classici ed euristici k-means e gerarchici, concentrandosi sui modelli Bayesiani non parametrici, che rappresentano un approccio più flessibile e utile nel superamento di alcuni limiti rispetto ai primi.
In particolare, lo studio si concentra sull'utilizzo di modelli mistura, in cui ogni componente corrisponde ad un cluster. I principali problemi sono quindi la determinazione del numero delle componenti e la distribuzione di probabilità di queste. In ambito classico, i parametri della mistura vengono tipicamente stimati grazie all'algoritmo expectation-maximization (EM) e le osservazioni vengono assegnate ai cluster sulla base della probabilità a posteriori di appartenere alla rispettiva componente.
Questo approccio presenta però due limiti: assume che il numero di componenti sia finito e che la misura dell'incertezza, rappresentata dalla probabilità a posteriori, non tenga conto dell'incertezza nelle stime dei parametri. Invece, i modelli non parametrici assumono che il numero di componenti della mistura sia infinito e di conseguenza consentono al numero di cluster di crescere con la raccolta di nuovi dati.

Così come in ambito Bayesiano classico, anche in quello non parametrico occorre stimare la probabilità a posteriori delle osservazioni di appartenere alle componenti e per questo si ricorre usualmente alle tecniche di Monte Carlo Markov Chain (MCMC). L'algoritmo MCMC genera ad ogni iterazione una partizione dei dati, che rappresentano campioni approssimati della posterior.
Il principale problema dei modelli non parametrici è rappresentato dall'elevata dimensionalità dello spazio delle partizioni, che cresce all'aumentare dei dati.
A causa dell'elevata dimensionalità dello spazio, l'MCMC genererà infatti un elevato numero di differenti partizioni anche simili tra loro, che differiscono solo per piccole variazioni ed è per questo che risulta necessario ricorrere a degli indicatori di sintesi.

In conclusione, ricorrendo alla teoria delle decisioni, la stima puntuale della distribuzione a posteriori corrisponde a quella che genera la partizione tale che la funzione di perdita attesa condizionata ai dati è minimizzata.

$$\mathbf{c}^* = \arg\min_{\hat{\mathbf{c}}} \mathbb{E}[L(\mathbf{c}, \hat{\mathbf{c}})|y_{1:N}]$$

L'idea è quindi quella di introdurre una funzione di perdita $L(\mathbf{c}, \hat{\mathbf{c}})$ che valuti il costo di un'errata assegnazione al cluster. La stima ottimale è quella che minimizza la perdita. Diventa allora fondamentale definire la funzione di perdita, che deve essere appropriata per lo spazio delle partizioni considerato.

Come alternativa alla stima puntuale, si possono utilizzare anche le "credible balls", che definiscono la regione delle partizioni con alta probabilità a posteriori e caratterizzano l'incertezza attorno alla stima puntuale.

2. Cenni al clustering Basyesiano non parametrico

Consideriamo i dati condizionatamente i.i.d. e con distribuzione

$$f(y|P) = \int K(y|\theta)\, dP(\theta)$$

dove $K(y|\theta)$ è una densità parametrica con parametro $\theta \in \Theta$ e P è una misura di probabilità su Θ. P richiede in generale una distribuzione a priori non parametrica che tipicamente ha realizzazioni discrete quasi certamente e si può pertanto scrivere

$$P = \sum_{j=1}^{\infty} w_j \delta_{\theta_j}$$

dove si assume che i pesi w_j e θ_j siano i.i.d. rispetto ad una misura base P_0. Quindi, si ottiene il modello mistura

$$f(y|P) = \sum_{j=1}^{\infty} w_j K(y|\theta_j).$$

Poichè, come già detto, P è discreto q.c., il modello genera una partizone latente c dei dati, in cui due punti appartengono allo stesso cluster se generati dalla stessa componente della mistura.
La partizione può essere rappresentata con $c = (C_1, ..., C_{k_N})$ dove C_j contiene gli indici dei punti contenuti nel j-esimo cluster e k_N è il numero di cluster nel campione di dimensione N.

Siano $y_j = \{y_n\}_{n \in C_j}$, la funzione di verosimiglianza dei dati condizionati alla partizione è

$$f(y_{1:N} \mid c) = \prod_{j=1}^{k_N} m(y_j) = \prod_{j=1}^{k_N} \int \prod_{n \in C_j} K(y_n \mid \theta) dP_0(\theta).$$

La distribuzione a posteriori della partizione è proporzionale al prodotto tra la prior e la verosimiglianza condizionata.

$$p(c \mid y_{1:N}) \propto p(c) \prod_{j=1}^{k_N} m(y_j),$$

dove la prior $p(c)$ è determinata dalla prior scelta come misura di probabilità della mistura. Nel nostro lavoro utilizziamo il Processo di Dirichlet e possiamo pertanto scrivere

$$p(c) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \alpha^{k_N} \prod_{j=1}^{k_N} \Gamma(n_j),$$

dove α è il parametro di concentrazione e n_j è la numerosità del j-esimo cluster.

3. Funzioni di perdita

Come già accennato in precedenza, le funzioni di perdita consentono di misurare il costo associato ad un'errata assegnazione ai cluster e sono fondamentali per la stima puntuale della partizione. Ricordiamo infatti che la stima c^* corrisponde alla partizone che minimizza il valore atteso della funzione di perdita a posteriori.
In questo elaborato presenteremo e confronteremo due funzioni di perdita.

3.1 Binder's Loss

La funzione di perdita espressa da Binder è spesso la più ampiamente utilizzata ed è

$$B(c, \hat{c}) = \sum_{i<j} aI(c_i = c_j)I(\hat{c}_i \neq \hat{c}_j) + bI(c_i \neq c_j)I(\hat{c}_i = \hat{c}_j),$$

dove $c = c_1, ..., c_n$ sono le vere etichette assegnate alle osservazioni e $\hat{c} = \hat{c_1}, ..., \hat{c_n}$ la loro stima. Invece, $a > 0$ e $b > 0$ rappresentano il costo assegnato a due tipi di errore. Binder infatti penalizza, per tutte le possibili coppie di osservazioni, l'errore di allocare due osservazioni in cluster differenti quando dovrebbero appartenere nello stesso (a) e l'errore di allocare allo stesso cluster due osservazioni che dovrebbero essere separate (b).

3.1.1 Versione n-invariante

Una versione n-invariante proposta da Wade e Gahahramani assume $a = b = 1$ ed è

$$B_{\text{n-inv}}(\rho, \hat{\rho}) = \frac{2}{n^2} B(\rho, \hat{\rho}) = \sum_{S \in \rho} \left(\frac{|S|}{n} \right)^2 + \sum_{\hat{S} \in \hat{\rho}} \left(\frac{|\hat{S}|}{n} \right)^2 - 2 \sum_{S \in \rho} \sum_{\hat{S} \in \hat{\rho}} \left(\frac{|S \cap \hat{S}|}{n} \right)^2 .$$

dove ρ rappresenta la partizone della popolazione e S il generico cluster.
In questo specifico caso, la partizione ottima c^* è la partizione che minimizza

$$\sum_{i<j} \left(I(c_i = c_j) - p_{ij} \right)^2$$

dove $p_{ij} = P(c_i = c_j \mid y_{1:N})$ è la probabilità a posteriori che le due generiche osservazioni i e j siano inserite nello stesso cluster.

Inoltre, il valore massimo assumibile da questa funzione di perdita è $1 - \frac{1}{N}$, cioè:

$$B_{\text{n-inv}}(\rho, \hat{\rho}) \leq 1 - \frac{1}{N}$$

La funzione di perdita Binder presenta un problema importante: tende a sovrastimare il numero dei cluster. Questo avviene perchè tende a separare due unità piuttosto che unirle. Questo problema può essere risolto pesando maggiormente l'errore che si commette separando due osservazioni che appartengono allo stesso cluster, ovvero occorre aumentare il costo rappresentato dal parametro a ed ottenere $a > b$.

3.1.2 Versione generalizzata

Per contrastare il problema della sovrastima del numero di cluster, Dahl et al. introducono una nuova versione della Binder n-invariante, utilizzando pesi a e b generici.

$$B_{\text{general}}(\rho, \hat{\rho}) = a \sum_{S \in \rho} \left(\frac{|S|}{n} \right)^2 + b \sum_{\hat{S} \in \hat{\rho}} \left(\frac{|\hat{S}|}{n} \right)^2 - (a+b) \sum_{S \in \rho} \sum_{\hat{S} \in \hat{\rho}} \left(\frac{|S \cap \hat{S}|}{n} \right)^2 .$$

dove, ponendo $a > b$, è possibile controllare il numero di cluster nella stima.

3.2 Variation of information

Un'alternativa alla Binder loss è la Variation of Information (VI), che si basa sull'idea di misurare la distanza tra due partizoni. La funzione, proposta da Wade e Ghahramani, è

$$\begin{aligned} VI(\rho, \hat{\rho}) &= H(\rho) + H(\hat{\rho}) - 2I(\rho, \hat{\rho}) \\ &= -H(\rho) - H(\hat{\rho}) + 2H(\rho, \hat{\rho}) \\ = \sum_{S \in \rho} \frac{|S|}{n} \log_2 \left(\frac{|S|}{n} \right) &+ \sum_{\hat{S} \in \hat{\rho}} \frac{|\hat{S}|}{n} \log_2 \left(\frac{|\hat{S}|}{n} \right) - 2 \sum_{S \in \rho} \sum_{\hat{S} \in \hat{\rho}} \frac{|S \cap \hat{S}|}{n} \log_2 \left(\frac{|S \cap \hat{S}|}{n} \right) \end{aligned}$$

dove $H(\rho)$ e $H(\hat{\rho})$ rappresentano l'entropia individuale, $H(\rho, \hat{\rho})$ l'entropia congiunta e $I(\rho, \hat{\rho})$ è l'informazione condivisa da ρ e $\hat{\rho}$.
Dato che $H(\rho)$ è una costante se consideriamo l'ottimizzazione rispetto a $\hat{\rho}$, si può concludere che nel caso della VI, la partizione ottima è data da

$$\hat{c}^* = \arg\min_{\hat{c}} \mathbb{E}(\text{VI}(c, \hat{c}) \mid y_{1:N})$$

$$\hat{c}^* = \arg\min_{\hat{c}} \sum_{i=1}^{n} \log_2 \left(\sum_{j=1}^{n} I(\hat{c}_j = \hat{c}_j) \right) - 2 \sum_{i=1}^{n} \mathbb{E} \left[\log_2 \left(\sum_{j=1}^{n} I(c_j = c_i) I(\hat{c}_j = \hat{c}_i) \right) \mid y_{1:N} \right].$$

Inoltre, il valore massimo assumibile da questa funzione di perdita è $log_2(N)$, cioè:

$$VI(\rho, \hat{\rho}) \leq log_2(N)$$

3.2.1 Versione generalizzata

Al contrario della funzione di perdita di Binder, la VI potrebbe sottostimare il numero di cluster reale e pertanto, anche in questo caso, si introduce una versione generalizzata.

$$\begin{aligned} VI_{general}(\rho, \hat{\rho}) &= bH(\rho) + aH(\hat{\rho}) - (a+b)I(\rho, \hat{\rho}) \\ &= -aH(\rho) - bH(\hat{\rho}) + (a+b)H(\rho, \hat{\rho}) \\ = a \sum_{S \in \rho} \frac{|S|}{n} \log_2 \left(\frac{|S|}{n} \right) &+ b \sum_{\hat{S} \in \hat{\rho}} \frac{|\hat{S}|}{n} \log_2 \left(\frac{|\hat{S}|}{n} \right) - (a+b) \sum_{S \in \rho} \sum_{\hat{S} \in \hat{\rho}} \frac{|S \cap \hat{S}|}{n} \log_2 \left(\frac{|S \cap \hat{S}|}{n} \right) \end{aligned}$$

dove $a, b > 0$ e contrario del caso precedente, il controllo da applicare è $a < b$.

Le funzioni di perdita presentate, ad eccezione delle forme generalizzate proposte da Dahl et al., soddisfano le seguenti proprietà e per questo possono essere definite metriche:

- Principio di identità degli indiscernibili
- Simmetria
- Disuguaglianza triangolare

3.3 Credible Ball

Poichè la Binder's loss e la VI sono definite metriche per come descritte da Wade e Ghahramani, è possibile costruire una Credible Ball attorno alla stima ottenuta.
La Credible Ball si basa sull'idea di costruire un intervallo sferico di livello 1 - α attorno alla stima puntuale c^*, con $\alpha \in [0, 1]$:

$$B_{\epsilon^*}(c^*) = \{c : d(c^*, c) \leq \epsilon^*\}$$

dove ϵ^* è il più piccolo $\epsilon > 0$ tale che $P(B_\epsilon(c^*)|y_{1:N}) \geq 1 - \alpha$. Si tratta dunque della più piccola sfera attorno a c^* con probabilità a posteriori pari o superiore a $1-\alpha$ e questo descrive l'incertezza a posteriori nella stima puntuale.

4. Algoritmi: Greedy e SALSO

Procediamo ora con la descrizione dei due algoritmi che utilizzeremo nella fase simulativa: il Greedy Algorithm (Wade e Ghahramani, 2018) e il SALSO Algorithm (Dahl, Johnson e Muller, 2021).
Entrambi i metodi appartengono alla famiglia degli algoritmi greedy, i quali, per definizione, ad ogni iterazione eseguono dei piccoli aggiornamenti localmente ottimali lungo la strada per trovare la soluzione finale.
Essi si pongono l'obiettivo di trovare la miglior stima della partizione, e quindi il miglior clustering dei dati, a partire dai campioni generati da MCMC. Essendo greedy algorithms non si limitano soltanto alle partizioni visitate dalla catena, ma possono esplorarne di nuove. Questo risulta molto importante perchè, in quasi tutti gli esempi simulati e reali, il clustering stimato non è tra le partizioni campionate e dunque si può ottenere una perdita attesa inferiore.

4.1 Greedy Search Algorithm

Il Greedy Search Algorithm, proposto da Wade e Ghahramani, parte da una partizione iniziale ($\hat{c}$) che può essere selezionata tra quelle campionate dalla catena MCMC o utilizzando un algoritmo di clustering gerarchico.
Dopodichè effettua passi localmente ottimali in un intorno di partizioni definite rispetto alla funzione di perdita scelta e al diagramma di Hasse; una struttura che rappresenta lo spazio delle partizioni come un reticolo in cui i nodi sono le tutte le possibili partizioni, mentre gli archi collegano le partizioni che differiscono per un singolo elemento.
In pratica, l'algoritmo esplora un numero limitato di partizioni vicine, definite dalla dimensione dell'intorno (l).
Per ogni partizione vicina viene calcolata la perdita attesa a posteriori. Questo può essere complesso, ma Wade e Ghahramani propongono un metodo per semplificare il calcolo sfruttando la disuguaglianza di Jensen e lavorando con un limite inferiore della perdita attesa.
Infine, viene selezionata la partizione che minimizza la perdita attesa a posteriori.
Si itera il processo e l'algoritmo termina quando non si ottengono più riduzioni significative della perdita attesa a posteriori o quando viene raggiunto il numero massimo di iterazioni pre impostato.
E' importante sottolineare che questo algoritmo può essere eseguito solo per la Binder's loss e il limite inferiore della VI e non per le loro versioni generalizzate. Questo è dovuto al fatto che l'implementazione prevede l'utilizzo della stima della matrice di similarità a posteriori.
Inolte, il risultato dell'algoritmo è sensibile allo starting value $\hat{c}$ e allo step size l, perchè rischia di rimanere intrappolato in minimi locali quando attraversa il diagramma di Hasse. Per mitigare questo problema gli autori consigliano di implementare inizializzazioni multiple, ad esempio in corrispondenza di diversi campioni MCMC o della migliore partizione trovata con altri algoritmi di ricerca.
Al contempo, un valore più grande di l garantisce una maggiore esplorazione delle partizioni riducendo la necessità dei restart multipli, ma questo può essere molto oneroso a livello computazionale se il dataset è di grande dimensioni.

Infatti, la complessità computazionale di questo algoritmo è $O(lN^2)$, dove l rappresenta il numero di partizioni da considerare ad ogni iterazione, mentre N è il numero di unità considerate.

4.2 SALSO: Sequentially-Allocated Latent Structure Optimization

L'algoritmo Sequentially-Allocated Latent Structure Optimization (SALSO), proposto da Dahl et al. è un algoritmo di ricerca greedy stocastico ed è composto da 4 fasi.

1. **Inizializzazione:** la partizione può essere inizializzata in due modi;

   1.a **Assegnazione Sequenziale:** le unità vengono allocate una alla volta in un cluster esistente o in uno nuovo, scegliendo l'allocazione che minimizza la stima Monte Carlo della perdita attesa a posteriori e ignorando ogni altra unità che deve essere ancora allocata. L'ordine in cui le unità vengono considerate è determinato da una permutazione casuale degli indici campionata in modo uniforme. Il numero di cluster iniziali può crescere fino a k_d, che indica il numero massimo di cluster desiderato.

   1.b **Assegnazione Casuale:** viene generata una partizione casuale tramite le etichette dei cluster, che sono ottenute campionando uniformemente le labels $1, ..., k_d$.

   La scelta tra i due metodi di inizializzazione è determinata da una variabile casuale uniforme, con una probabilità di allocazione sequenziale, indicata con p_{SA}, che può essere specificata dall'utente.

2. **Sweetening:** questo processo è simile all'allocazione sequenziale (1.a), ma in questo caso si ha un processo iterativo e non "one shot". Esso consiste nel riallocare casualmente le unità, una alla volta, in un ordine casuale. In particolare, tutte le unità sono allocate e ognuna di esse, una alla volta nell'ordine determinato da una permutazione campionata uniformemente tra tutte le possibili, è rimossa dal suo cluster e riallocata in un cluster esistente o in uno nuovo in base alla scelta che minimizza la stima Monte Carlo della perdita attesa a posteriori. Si ripete il processo fino a quando non avviene nessun cambiamento dopo un passaggio completo attraverso tutte le n unità.

3. **Zealous Update:** in questa fase l'algoritmo cerca di abbandonare i minimi locali introducendo un elemento di distruzione e ricostruzione casuale dei cluster per superare i limiti della ricerca greedy nelle fasi precedenti.
   In particolare, un numero limitato di cluster ($n_{maxZealous}$) vengono selezionati casualmente e "distrutti", ovvero tutte le unità che lo compongono vengono rimosse. Successivamente, gli elementi vengono riassegnati ai cluster esistenti o a nuovi cluster, sempre cercando di minimizzare la perdita attesa a posteriori. Se la partizione risultante ha una perdita attesa a posteriori inferiore a quella precedente, l'aggiornamento viene mantenuto; altrimenti, la partizione viene ripristinata allo stato precedente.

4. **Recording:** la stima Monte Carlo della perdita attesa a posteriori viene registrata per lo stato corrente.

L'algoritmo è implementato in R per svariate funzioni di perdita, tra cui quelle descritte nella sezione 3.

La complessità computazionale di questo algoritmo è $O(Hk_dk_hN)$, dove H è il numero di campioni MCMC, k_d è il numero massimo di cluster desiderati, k_h è il numero di cluster osservati e N è il numero di unità.
Nonostante ciò, l'algoritmo risulta estremamente efficiente grazie ad alcuni accorgimenti computazionali che sono stati eseguiti dagli autori.

SALSO viene definito "imbarazzamente parallelo" perchè consente di eseguire più run simultaneamente, sfruttando le risorse multi-core e riducendo i tempi di calcolo. Infatti, come per il greedy algorithm, anche qui si consiglia di effettuare ripetizioni multiple per ottenere una migliore stima del clustering, ma grazie a questa caratteristica non risulta eccessivamente oneroso. Inoltre, gli autori hanno implementato degli shortcut per il calcolo della perdita attesa a posteriori e la possibilità di selezionare il numero massimo di cluster desiderati, garantendo così una maggiora efficienza computazionale complessiva.

5. Studio di simulazione

In questa sezione eseguiamo uno studio di simulazione per fare un confronto tra le funzioni di perdita e gli algoritmi descritti nelle sezioni precedenti.

Simuliamo un dataset di $n = 100$ osservazioni da una mistura di 4 normali univariate

$$X_i \overset{iid}{\sim} \sum_{j=1}^{4} p_j N(\mu_j, \sigma_j)$$

dove $p = (0.2, 0.45, 0.25, 0.1)$, $\mu = (0, 3, 6, 9)$, $\sigma = (1, 0.5, 0.7, 0.3)$ e $i = 1, 100$.

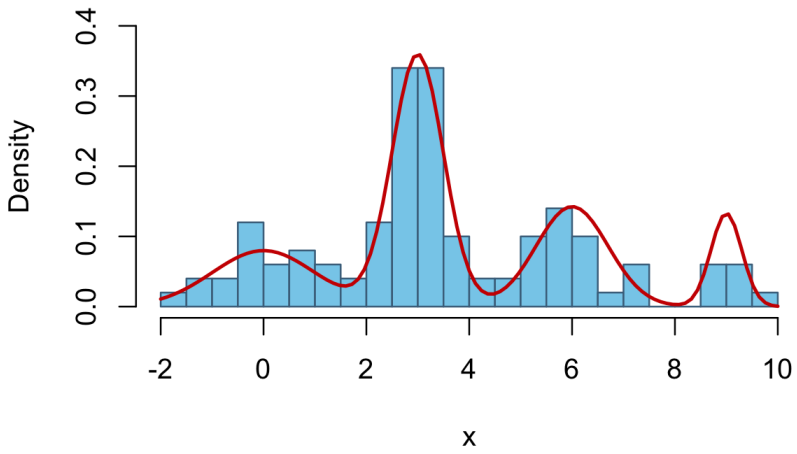

Figura 1: Istogramma dei dati generati dalla mistura con funzione di densità sovrapposta

Al fine di stimare la funzione di densità del dataset generato, utilizziamo un DPM "location" con kernel gaussiano univariato $K(x; \theta, \sigma^2)$, tale che:

$$K(X; \theta, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right)$$

e poniamo una prior $Inv - Gamma(1, 1)$ per il parametro σ^2.
Questo è un modello per osservazioni $X_i \in \mathbb{R}$, definito come $X_i|\tilde{P} \overset{iid}{\sim} \tilde{f}$, con $\tilde{P} \sim DP(\alpha, P_0)$, dove

$$\tilde{f}(x) = \int_{\mathbb{R}^+} \int_{\mathbb{R}} K(x; \theta, \sigma^2)\, d\tilde{P}(\theta)\, dIG(\sigma^2)$$

Riscrivendo il modello in forma gerarchica si ha:

$$X_i|\theta_i \overset{ind}{\sim} K(.; \theta_i, \sigma^2)$$

$$\theta_i|\tilde{P} \overset{iid}{\sim} \tilde{P}, \quad i = 1, \ldots, n$$
$$\sigma^2 \sim Inv - Gamma(1, 1)$$
$$\tilde{P} \sim DP(\alpha, P_0)$$

dove abbiamo introdotto un vettore di variabili aleatorie ausiliarie $\underline{\theta} = (\theta_1, ..., \theta_n)$.
Il nostro obiettivo è stimare la PDF della distribuzione delle osservazioni $\underline{X} = (X_1, ..., X_n)$, tramite $\hat{f}(x) = \mathbb{E}[\tilde{f}(x)|\underline{X}]$.
Dunque stimiamo $\hat{f}$ via Monte Carlo. Chiamiamo $\{\underline{\theta}^{(1)}, ..., \underline{\theta}^{(M)}\}$, con $\underline{\theta}^{(m)} = (\theta_1^{(m)}, ..., \theta_n^{(m)})$ per ogni $m = 1, ..., M$, un insieme di M realizzazioni della distribuzione a posteriori di $\underline{\theta}$ condizionatamente a $\underline{X}$. Allora possiamo valutare $\hat{f}$ come

$$\hat{f}(x) \approx \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}[\tilde{f}(x)|\underline{X}, \underline{\theta}^{(m)}, \sigma^{2(m)}]$$

Per generare le realizzazioni $\{\underline{\theta}^{(1)}, ..., \underline{\theta}^{(M)}\}$ ricorriamo ad un Gibbs sampling con 1200 iterazioni di cui 200 di burn-in (valori che suggeriscono convergenza valutando trace plot e acf).

Il clustering simulato dalla mistura, che da ora in poi definiremo come "clustering reale", è visibile in figura 2

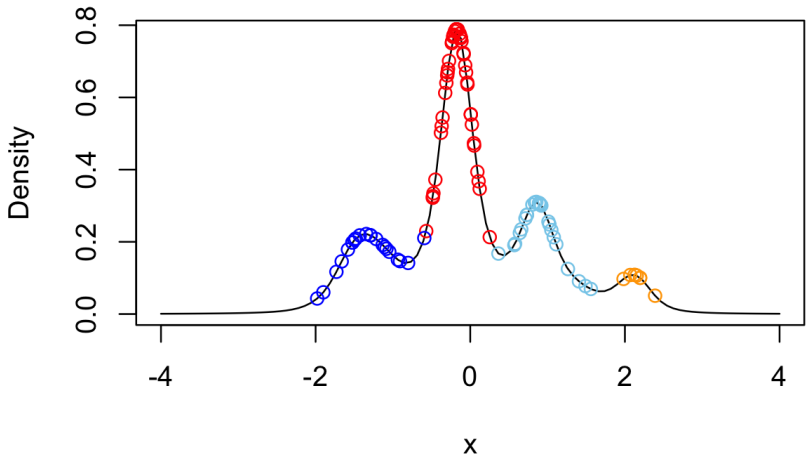

Figura 2: Rappresentazione dei 4 cluster reali - osservazioni colorate per appartenenza al cluster

Ora si vuole valutare la partizione ottima tra quelle campionate dopo il burn-in, utilizzando gli algoritmi descritti precedentemente e valutando i loro tempi di esecuzione.
Inoltre, confronteremo le diverse funzioni di perdita descritte, al fine di valutare la validità empirica della spiegazione teorica.

5.1 Risultati Greedy Algorithm

Osservando i grafici 3 e 4 e la tabella 1 possiamo notare che la Binder's loss tende, come previsto, a sovrastimare il numero di cluster, generandone $k_N^* = 12$, invece dei 4 reali. Di conseguenza, stima nel modo errato il 18% delle osservazioni.
La VI, invece, riesce a identificare il numero di cluster corretto e fornisce una quasi perfetta assegnazione delle osservazioni, precisamente del 98%.
Nonostante ciò, la perdita attesa della Binder's loss risulta essere leggermente inferiore.
Infine, il tempo di esecuzione è molto elevato per entrambe le funzioni di perdita, ma la minimizzazione della Binder's Loss, rispetto alla VI, sembra far essere più efficiente l'algoritmo a livello computazionale.

Loss	k_N^*	N_i	$\mathbb{E}(L\|x)$	$time_{sec}$
Binder	12	18	0.162	17.823
VI	4	2	0.173	32.962

Tabella 1: Greedy algorithm - risultati per il confronto delle funzioni di perdita Binder e VI in termini di 1) numero di cluster stimati k_N^*; 2) numero delle osservazioni classificate erroneamente N_i; 3) perdita attesa a posteriori normalizzata rispetto al suo valore massimo; 4) tempo di esecuzione in secondi

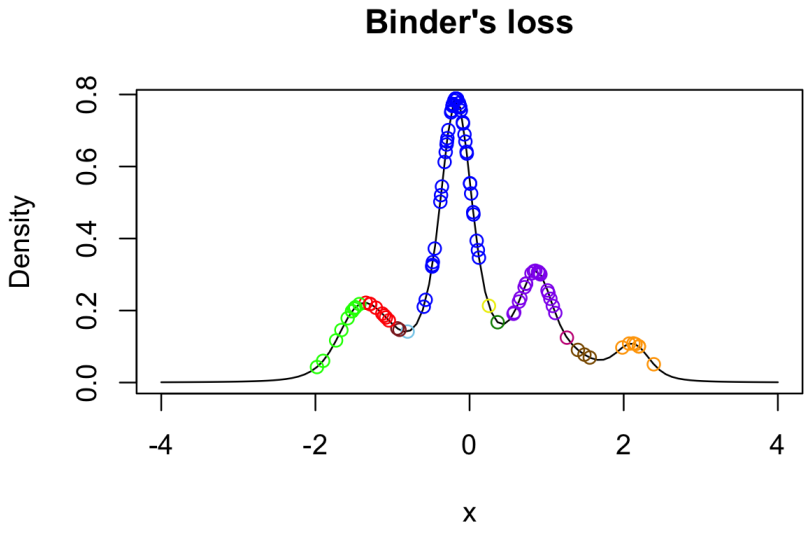

Figura 3: Rappresentazione del clustering con Binder's loss - 12 cluster - osservazioni colorate per appartenenza al cluster

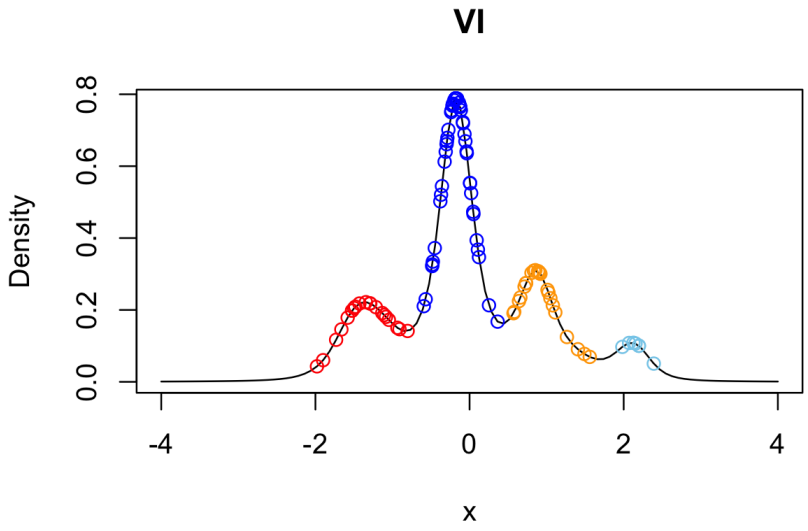

Figura 4: Rappresentazione del clustering con VI - 4 cluster - osservazioni colorate per appartenenza al cluster

5.1.1 Credible Ball

Per questo algoritmo è implementata la possibilità di calcolare la stima dell'incertezza del clustering utilizzando le credible ball.
La credible ball è riassunta attraverso i limiti verticali superiori (upper bounds), verticali inferiori (lower bounds) e orizzontali (horizontal bounds), definiti rispettivamente come le partizioni nella credible ball con il minor numero di cluster che sono più distanti da c^* (stima del clustering), con il maggior numero di cluster che sono più distanti da c^*, e con la maggiore distanza da c^*.

Nella tabella 2 sono riassunti i risultati ottenuti e da questi possiamo dedurre che, indipendentemente dalla metrica usata, si può notare una forte variabilità intorno alla partizione ottima. Infatti, il numero di cluster non rimane costante nei 3 bounds e di conseguenza possiamo affermare che è presente incertezza sul come le osservazioni vengono clusterizzate. Inoltre, i valori di $d(c*, c_{u/l/h})$ mostrano che le partizioni nei limiti sono effettivamente diverse dalla stima puntuale. La distanza è ovviamente influenzata dalla metrica scelta, ma possiamo notare come la distanza più ampia sia maggiore nel limite orizzontale per entrambe le funzioni di perdita.

	Upper		Lower		Horizontal	
Loss	k_N^u	$d(c*, c_u)$	k_N^l	$d(c*, c_l)$	k_N^h	$d(c*, c_h)$
Binder	4	0.161	16	0.172	8	0.362
VI	2	2.616	16	1.745	5	3.065

Tabella 2: Sintesi dei credible bounds (con livello di significatività credible ball del 95%) per Binder's loss e VI in termini di numero di cluster (k_N) e distanza della stima del cluster per il limite superiore, inferiore e orizzontale.

5.2 Risultati SALSO

Ora analizziamo i risultati forniti dall'algoritmo SALSO per cui sono implementate anche le funzioni di perdita generalizzate. Come detto in teoria, il poter variare a garantisce maggiore flessibilità nella creazione dei cluster. Di conseguenza, generiamo il clustering ponendo $a = 1$ per applicare le funzioni classiche, ma cerchiamo anche il valore di a che garantisca il numero di cluster esatto e minimizzi il numero di unità misclassificate.

I risultati descritti in tabella 3 e la visualizzazione dei grafici 5, 6, 7 e 8 ci confermano che la Binder's loss classica tende a sovrastimare il numero di cluster, mentre riusciamo a ottenere il numero di cluster corretto ponendo $a = 1.6$. Questa generalizzazione consente, inoltre, di ridurre le unità misclassificate da 12 a 4.
La VI standard, invece, sovrastima il numero di cluster di un solo elemento, generando 5 osservazioni classificate erroneamente; dunque una piccola variazione di a, in particolare pari ad $a = 1.1$, ci porta ad ottenere esattamente 4 cluster e ad un'accuracy del 98%.
Anche la perdita attesa si riduce per entrambe le funzioni di perdita nel momento in cui si considera la loro versione generalizzata.
Infine, confermiamo quanto spiegato precedentemente: il SALSO garantisce un'efficienza computazionale imparagonabile al Greedy algorithm, riducendo i tempi di esecuzione drasticamente.

Loss	k_N^*	N_i	$\mathbb{E}(L\|x)$	$time_{sec}$
Binder $a = 1$	12	17	0.163	0.071
Binder $a = 1.6$	4	2	0.115	0.026
VI $a = 1$	5	5	1.444	0.029
VI $a = 1.1$	4	2	1.365	0.025

Tabella 3: SALSO - risultati per il confronto delle funzioni di perdita Binder e VI in termini di 1) numero di cluster stimati k_N^*; 2) numero delle osservazioni classificate erroneamente N_i; 3) perdita attesa a posteriori; 4) tempo di esecuzione in secondi

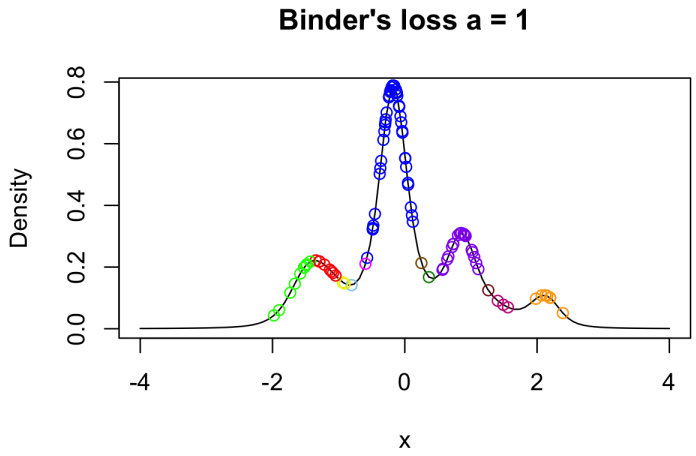

Figura 5: Rappresentazione del clustering stimato con Binder's loss, $a = 1$ - 12 cluster - osservazioni colorate per appartenenza al cluster

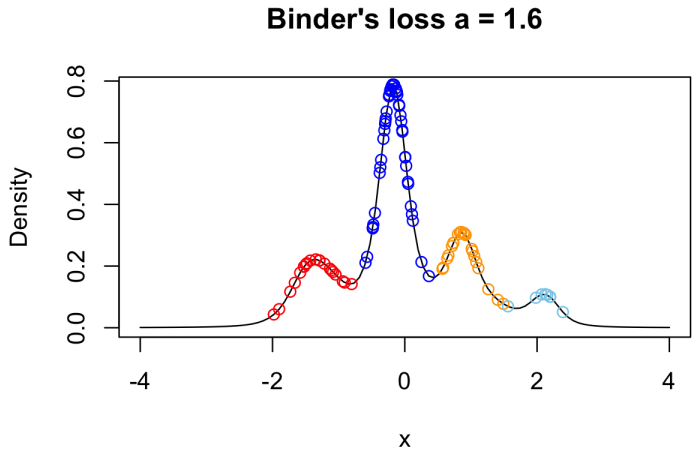

Figura 6: Rappresentazione del clustering stimato con Binder's loss generalizzata, $a = 1.6$ - 4 cluster - osservazioni colorate per appartenenza al cluster

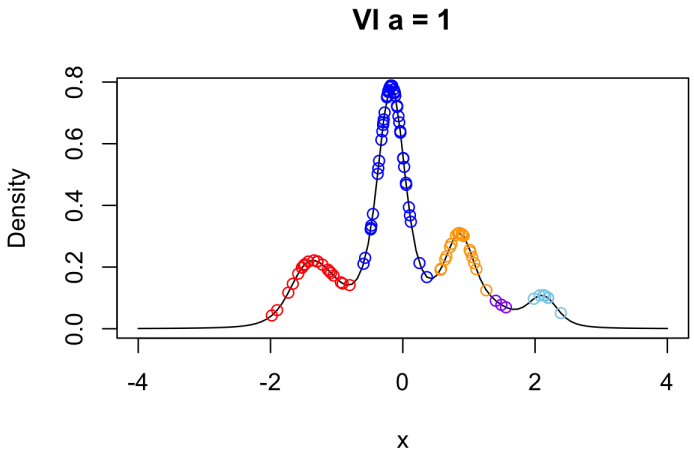

Figura 7: Rappresentazione del clustering stimato con VI, $a = 1$ - 5 cluster - osservazioni colorate per appartenenza al cluster

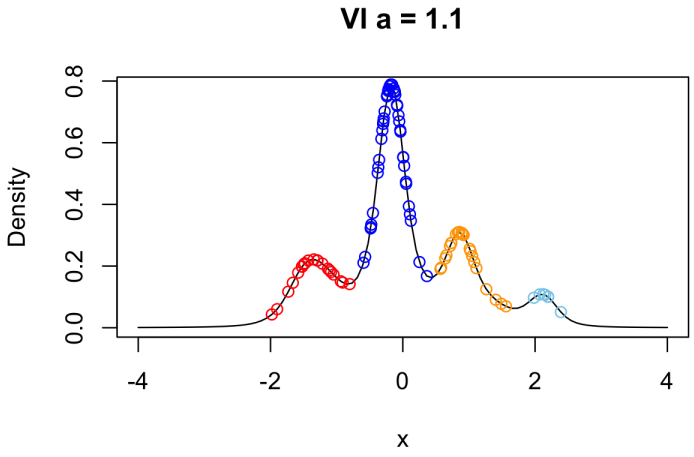

Figura 8: Rappresentazione del clustering stimato con VI, $a = 1.1$ - 4 cluster - osservazioni colorate per appartenenza al cluster

6. Analisi di un dataset reale

Concludiamo lo studio con l'analisi di un dataset reale, provando a partizionarne le istanze tramite i metodi appena discussi e applicati nell'esercizio simulativo. Il dataset che abbiamo utilizzato è "Country Data", disponibile sulla piattaforma Kaggle.
Il dataset si compone di 167 osservazioni e 10 variabili, le osservazioni sono determinate dai Paesi e le variabili, di carattere economico oppure medico, sono di tipo numerico e descrivono lo sviluppo socio-economico dei Paesi. Dopo aver studiato le variabili e aver determinato quali sono le maggiormente significative, ci limitiamo all'analisi delle seguenti.

Variabile	**Descrizione**
Export	Esportazioni di beni e servizi pro capite, espresse come % del PIL pro capite.
Import	Importazioni di beni e servizi pro capite, espresse come % del PIL pro capite.
Gdpp	PIL pro capite, calcolato come il PIL totale diviso per la popolazione totale.
Life Expectancy	Numero medio di anni di vita se gli attuali modelli di mortalità rimanessero invariati.
Total Fertility	Numero di figli per ogni donna se i tassi di fertilità per età attuali rimanessero invariati.

Tabella 4: Descrizione delle variabili del dataset "Country Data"

L'obiettivo dell'applicazione è quello di categorizzare i Paesi utilizzando fattori socio-economici e sanitari che ne determinano lo sviluppo complessivo, al fine di stabilire quali siano le realtà maggiormente bisognose e quali invece siano particolarmente sviluppate.

Nell'applicazione, abbiamo deciso di utilizzare la funzione di perdita Binder generalizzata e quindi applicare l'algoritmo SALSO singolarmente per ognuna delle cinque variabili descritte sopra.
Per ogni variabile abbiamo modificato il parametro a al fine di ottenere sempre quattro cluster, così da rendere i risultati interpretabili.

6.1 Interpretazione dei risultati

Una volta ottenuti i cluster, abbiamo determinato quale fosse il livello assunto dalle variabili in ogni Paese: basso, medio, alto o molto alto. In seguito, consultando le informazioni pubblicate dal Ministero degli Esteri, abbiamo trovato un riscontro reale dei risultati ottenuti ed una spiegazione socio-economica.

Come visibile dai grafici 9 e 10, la variabile **Gdpp**, che descrive il PIL pro capite, è di facile interpretazione e rispecchia, in linea generale, le dinamiche socio-economiche ampiamente documentate nella letteratura. Si registra infatti un Gdpp basso nella totalità dei Paesi del continente africano e in molti Paesi dell'America del Sud e dell'Asia Centrale. La quasi totalità dei Paesi europei e l'America del Nord presentano invece un Gdpp alto. Lussemburgo, Norvegia, Svizzera e Qatar spiccano per i valori molto alti.

Per quanto riguarda le variabili economiche, meno intuitiva è l'interpretazione delle variabili Import ed Export (espresse come % del PIL pro capite). Potremmo infatti erroneamente pensare che i Paesi maggiormente economicamente sviluppati e globalizzati siano quelli per i quali si registrano i maggiori livelli di importazioni ed esportazioni.
Focalizzandoci sui Paesi che registrano un alto livello di **Exports**, visualizzabile nei grafici 11 e 12, alcuni tra i risultati più sorprendenti sono:
Angola, Libia, Qatar, Kuwait e Guinea Equatoriale: il livello alto è dovuto alle loro abbondanti risorse naturali, in particolare petrolio e gas naturale.
Vietnam, Panama e Malesia: sono economie in rapido sviluppo, con una forte espansione delle esportazioni industriali e dei beni manifatturieri, come elettronica, abbigliamento e prodotti agricoli.
Singapore, Estonia, Irlanda, Lituania e Slovenia: hanno economie fortemente orientate ai servizi, tra cui quelli finanziari e tecnologici.

Analizzando invece la variabile **Import**, visualizzabile nei grafici 13 e 14, si nota che Paesi molto sviluppati come quelli europei o il Nord America hanno un livello di Import basso. Questo risultato denota come alcuni Paesi tendano a importare meno in percentuale del loro PIL pro capite grazie a una forte produzione interna di beni e servizi, politiche favorevoli all'industria locale e una specializzazione in settori ad alto valore aggiunto. Inoltre, concentrandosi maggiormente sulla produzione di beni e servizi ad alto valore, la necessità di importazioni fisiche è ridotta.

Analizziamo ora le variabili che descrivono il livello medico-sanitario.
La variabile **Life Expectancy**, osservabile nei grafici 15 e 16, esprime l'aspettativa di vita e, come è normale aspettarsi, si nota che i Paesi più sviluppati hanno un'aspettativa di vita maggiore rispetto a molti Paesi dell'Africa o di altre regioni mediamente o sottosviluppate.
I Paesi che registrano un livello particolarmente basso sono la Repubblica Centrafricana, Haiti e Lesotho. In tutti e tre i casi la motivazione principale è lo scarsissimo livello medico-sanitario. A conferma di ciò, si consideri che il Lesotho ha un tasso di prevalenza dell'HIV di oltre il 20% ed è la nazione con il più alto tasso di incidenza della tubercolosi del mondo.
Nella Repubblica Centrafricana la malaria e la lebbra sono ampiamente diffuse e il tasso di mortalità infantile è tra i più elevati. Anche ad Haiti molte malattie come la malaria e la tubercolosi sono diffuse. Inoltre, nel 2010, anno di riferimento del dataset, lo Stato ha subito un violento terremoto che ha ulteriormente abbassato il dato a soli 32 anni.

La variabile **Total Fertility** indica il numero medio di figli per ogni donna ed è visibile nei grafici 17 e 18. Gli Stati con tassi di fertilità molto alti presentano questa caratteristica a causa di scarsa istruzione femminile, limitato accesso alla contraccezione e cultura tradizionale che valorizza le famiglie numerose. Inoltre, l'alta mortalità infantile e la dipendenza economica dai figli rafforzano la necessità di avere più bambini.
La maggior parte dei Paesi, in particolar modo quelli sviluppati, sono raggruppati nel cluster dove il livello è basso. Questo è dovuto a urbanizzazione e ad alti costi di vita, migliore istruzione, accesso alla contraccezione e cambiamenti culturali che privilegiano famiglie più piccole. Le donne tendono a posticipare la maternità e avere meno figli, bilanciando carriera e vita familiare.

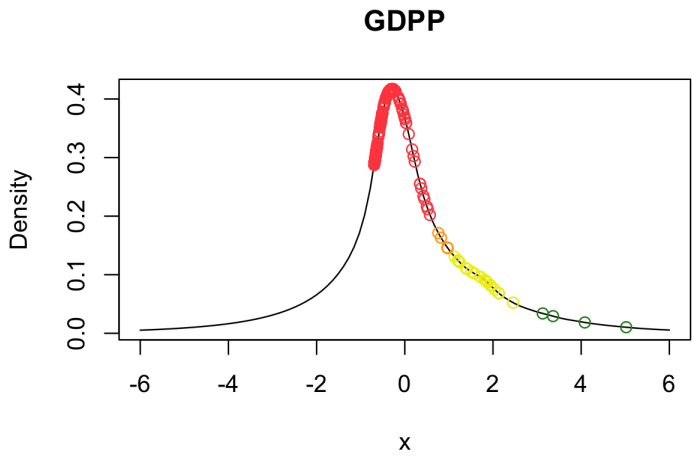

Figura 9: Gdpp, rappresentazione dei 4 cluster stimati - osservazioni colorate per appartenenza al cluster

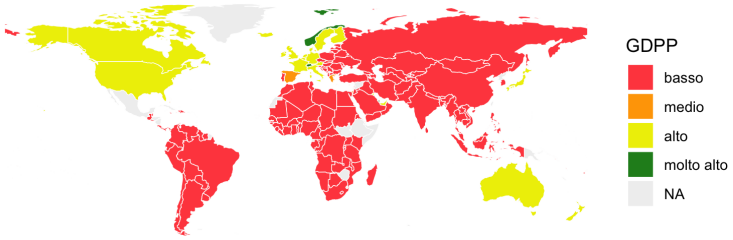

Figura 10: Gdpp, heatmap dei 4 cluster stimati - nazioni colorate per appartenenza al cluster

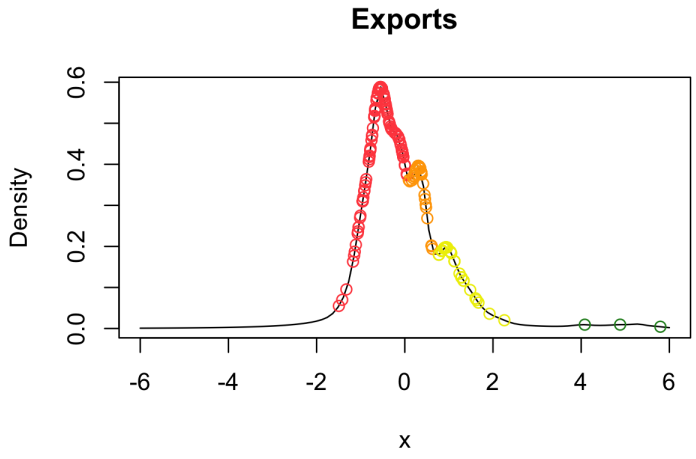

Figura 11: Exports, rappresentazione dei 4 cluster stimati - osservazioni colorate per appartenenza al cluster

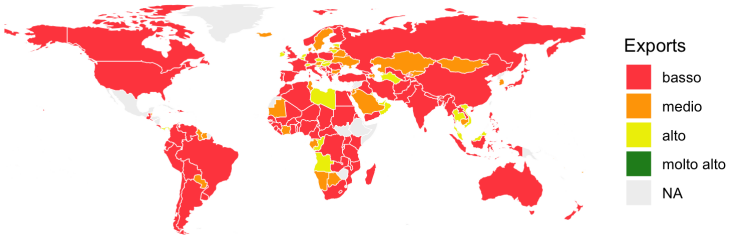

Figura 12: Exports, heatmap dei 4 cluster stimati - nazioni colorate per appartenenza al cluster

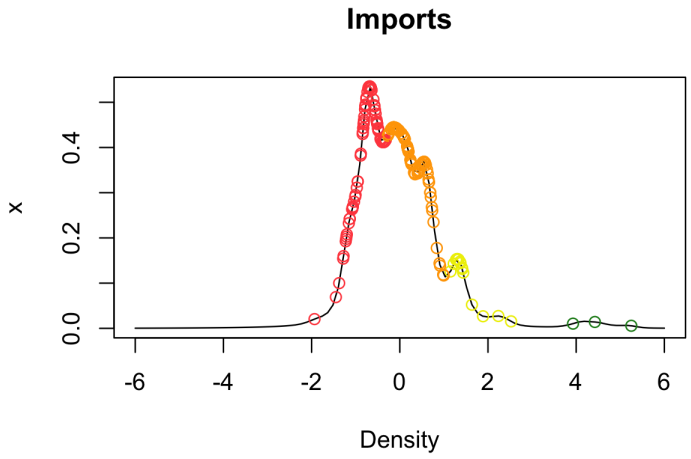

Figura 13: Imports, rappresentazione dei 4 cluster stimati - osservazioni colorate per appartenenza al cluster

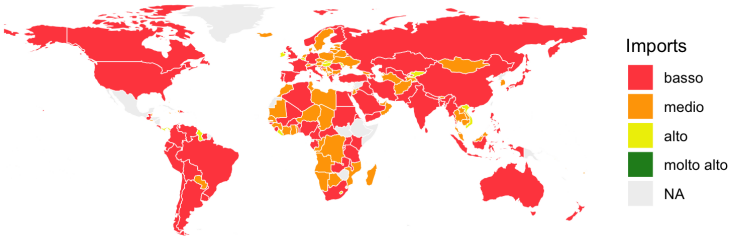

Figura 14: Imports, heatmap dei 4 cluster stimati - nazioni colorate per appartenenza al cluster

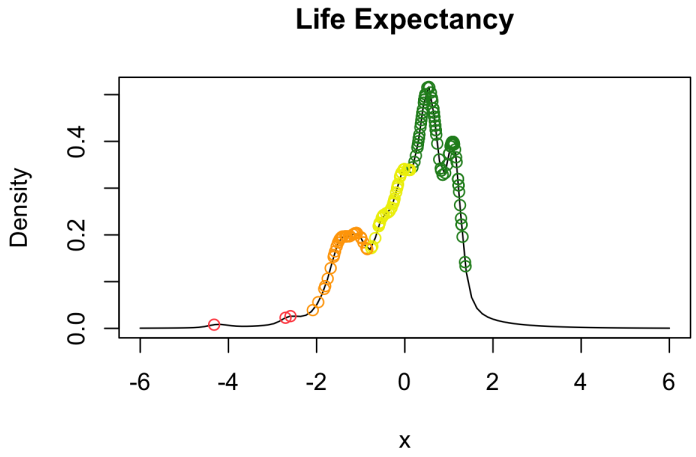

Figura 15: Life Expectancy, rappresentazione dei 4 cluster stimati - osservazioni colorate per appartenenza al cluster

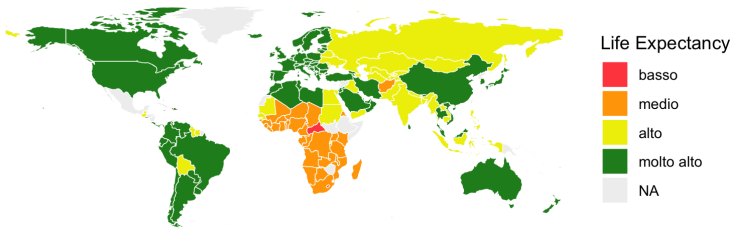

Figura 16: Life Expectancy, heatmap dei 4 cluster stimati - nazioni colorate per appartenenza al cluster

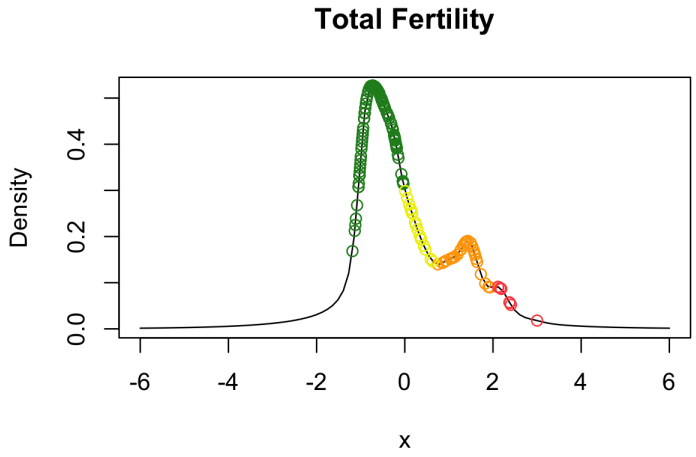

Figura 17: Total Fertility, rappresentazione dei 4 cluster stimati - osservazioni colorate per appartenenza al cluster

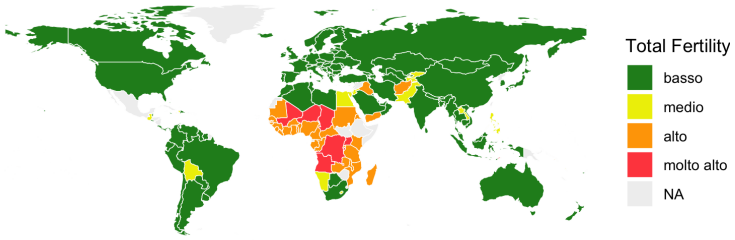

Figura 18: Total Fertility, heatmap dei 4 cluster stimati - nazioni colorate per appartenenza al cluster

6.2 Conclusioni

In conclusione, possiamo osservare che le variabili Import ed Export sono complesse nella loro interpretazione, poiché non riflettono il solo livello di sviluppo economico ma dinamiche più complesse, legate al modello produttivo di ciascun Paese. Ad esempio, nazioni come il Qatar o l'Angola, che hanno elevati livelli di esportazioni, spesso dipendono dalle risorse naturali come il petrolio. Altri paesi, come il Vietnam o la Malesia, mostrano economie in rapida espansione grazie a esportazioni manifatturiere che sfruttano però manodopera a basso costo.

Per quanto riguarda le variabili di tipo medico-sanitario, possiamo concludere che il divario tra Nord e Sud del mondo è molto marcato. Nei Paesi più sviluppati, l'aspettativa di vita è elevata grazie a migliori infrastrutture sanitarie, accesso ai farmaci, prevenzione e controllo delle malattie croniche. Al contrario, molti Paesi del Sud del mondo, specialmente nell'Africa sub-sahariana, registrano aspettative di vita più basse, spesso a causa di malattie trasmissibili come l'HIV, la malaria e la tubercolosi, oltre alla carenza di risorse mediche.

Il PIL pro capite è strettamente correlato alla qualità della sanità: i cluster individuati per questa variabile e per le variabili sanitarie sono molto simili. Paesi con un PIL pro capite elevato tendono a investire di più in infrastrutture sanitarie e nella formazione del personale medico, migliorando l'accesso ai servizi sanitari e la prevenzione delle malattie. Al contrario, nei Paesi con PIL pro capite basso, le risorse limitate spesso impediscono la creazione di sistemi sanitari efficienti.

L'analisi effettuata, in conclusione, ha evidenziato l'enorme divario e squilibrio globale. Le nazioni sviluppate presentano indicatori di qualità della vita decisamente buoni, ma diametralmente opposta è la situazione dei Paesi a basso reddito. Infatti, nonostante questi Paesi registrino livelli molto elevati di importazioni ed esportazioni, il loro coinvolgimento nell'economia globale sembra perlopiù essere il risultato di meccanismi di sfruttamento.

Riferimenti bibliografici

[1] S. Wade and Z. Ghahramani, *Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion)*. In: *Bayesian Analysis* (2018).

[2] David B. Dahl, Devin J. Johnson e Peter Müller. *Search algorithms and loss functions for Bayesian clustering*. In: Journal of Computational and Graphical (2022).

[3] S. Wade, Package: *mcclust.ext* (2015).

[4] David B. Dahl, Package: *salso* (2024).

[5] Kaggle, *Unsupervised Learning on Country Data* (2020)

[6] Farnesina, *Ministero degli Affari Esteri e della Cooperazione Internazionale* (2024)

[7] Indexmundi, *Dati storici* (2024)